09-16-05

AF
IFW

EXPRESS MAIL MAILING LABEL NO. ED 802269137 US

PATENT

In re Application of:
Frank S. Caccavale

Serial No.: 09/804,320
Confim. No. 1069

Filed: March 12, 2001

For: USING A VIRUS CHECKER IN ONE FILE SERVER TO CHECK FOR VIRUSES IN ANOTHER FILE SERVER

Group Art Unit: 2135

Examiner: Truong, Thanhnga B.

Atty. Dkt. No.: 10830.0075.NPUS00

APPEAL BRIEF TO THE BOARD OF PATENT APPEALS AND INTERFERENCES

Commissioner for Patents
PO Box 1450
Alexandria, Virginia 22313-1450

Sir:

This brief is in support of the appeal filed July 14, 2005, from the decision of the Examiner in the Final Official Action dated May 19, 2005. Please deduct the $500 fee of 37 C.F.R. 41.20(b)(2) for filing of the appeal brief from EMC Corporation Deposit Account No. 05-0889, Order No. EMC-00-172. A Fee transmittal form is enclosed for this purpose.

I. REAL PARTY IN INTEREST

The real party in interest is EMC Corporation, by virtue of an assignment recorded at Reel 011674 Frame 0819.

II. RELATED APPEALS AND INTERFERENCES

There are no related appeals or interferences.

III. STATUS OF THE CLAIMS

Claims 1 to 40 have been presented for examination.

Claims 1 to 40 have been finally rejected, and are being appealed.

IV. STATUS OF AMENDMENTS

No amendment was filed subsequent to the final rejection.

V. SUMMARY OF CLAIMED SUBJECT MATTER

The invention of appellant's claim 1 is a method of using a virus checker in one file server to check for viruses in another file server. (Appellant's title of the application.) A data processing system includes at least one client, a first file server coupled to the client for data access of the client to at least one file in the first file server, and at least a second file server coupled to the first file server for data access of the second file server to the file in the first file server. (Appellant's specification, page 4, lines 8-11.) The second file server is programmed with a virus checker program. (Appellant's specification, page 4, lines 11-12.) The virus checker program is executable by the second file server to perform an anti-virus scan upon file data in random access memory of the second file server. (Appellant's specification, page 4, lines 12-13.) The method includes the first file server responding to a request for access from the client to the file in the first file server by determining that an anti-virus scan of the file should be performed, and initiating the anti-virus scan of the file by sending to the second file server a request for the anti-virus scan including a specification of the file. (Appellant's specification,

page 4, lines 14-18.) Then the second file server responds to the request for the anti-virus scan by invoking the virus checker program to perform an anti-virus scan of the specified file by obtaining file data of the file from the first file server and storing the file data of the file into the random access memory of the second file server and performing the anti-virus scan upon the file data of the file in the random access memory. (Appellant's specification, page 4. lines 18-22.)

Appellant's FIG. 1 (reproduced below) shows a block diagram of a data processing system incorporating the invention. (Appellant's specification, page 12, lines 17-18.)

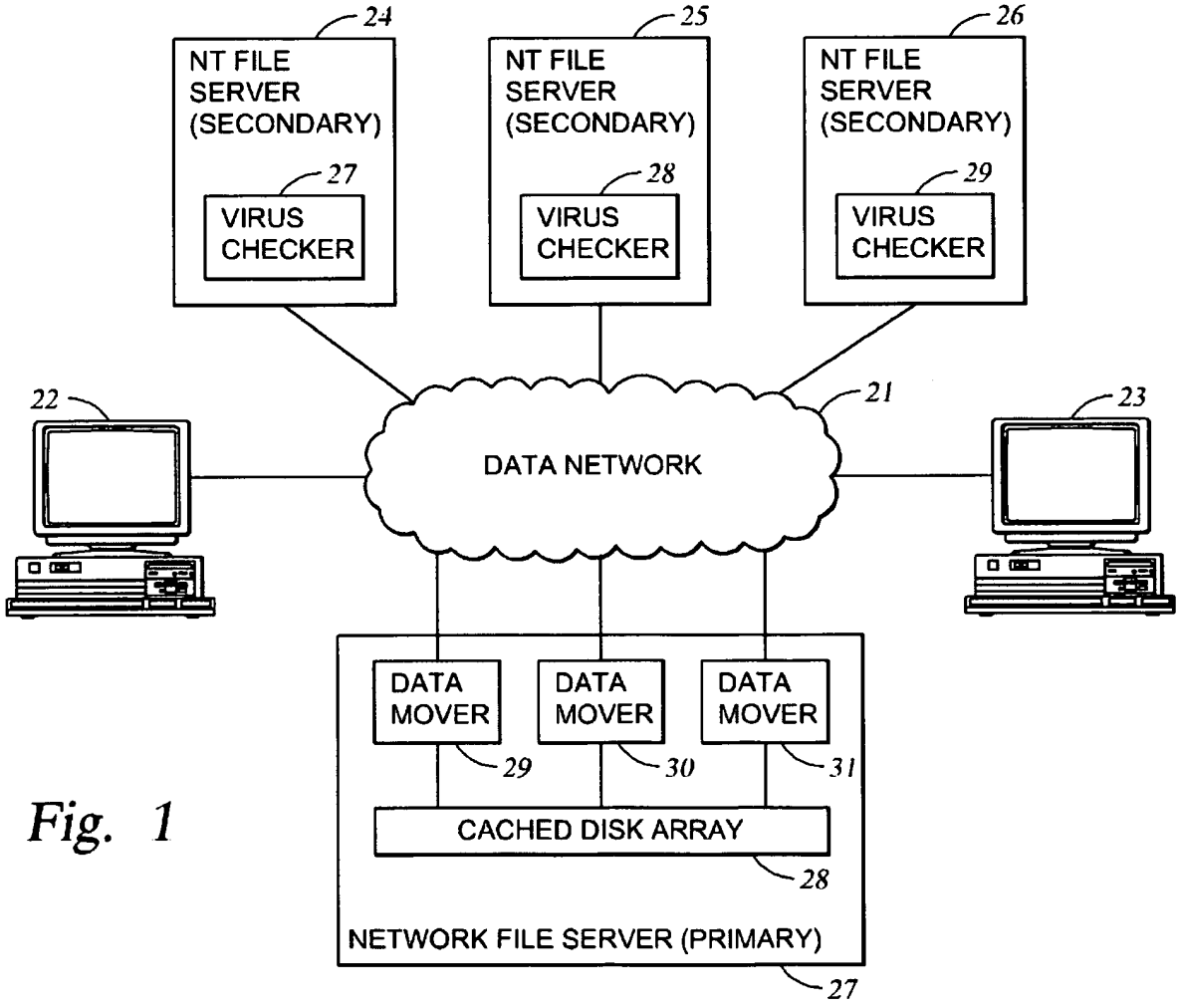

Fig. 1

The data processing system of FIG. 1 includes a data network 21 interconnecting a number of clients and servers. (Appellant's specification, page 14, lines 3-4.) The clients include work stations 22 and 23. The work stations, for example, are personal computers. The servers include conventional Windows NT/2000 file servers 24, 25, 26, and a very large capacity network file server 27. The network file server 27 functions as a primary server storing files in nonvolatile memory. The NT file servers 24, 25, 26 serve as secondary servers performing virus checking upon file data obtained from the network file server 27. (Appellant's specification, page 14, lines 6-11.)

Each of the NT file servers 24, 25, 26 is programmed with a respective conventional virus checker 27, 28, 29. The virus checkers are enterprise class anti-virus engines, such as the NAI/McAfee's NetShield 4.5 for NT Server, Symantec Norton AntiVirus 7.5 Corporate Edition for Windows NT, Trend Micro's ServerProtect 5.5 for Windows NT Server. In each of the NT file servers 24, 25, 26, the virus checker 27, 28, 29 is invoked to scan a file in the file server in response to certain file access operations. For example, when the file is opened for a user, the file is scanned prior to access by the user, and when the file is closed, the file is scanned before permitting any other user to access the file. (Appellant's specification, page 15, lines 7-15.)

The network file server 27, however, is not programmed with a conventional virus checker, because a conventional virus checker needs to run in the environment of a conventional operating system. Network administrators, who are the purchasers of the file servers, would like the network file server 27 to have a virus checking capability similar to the virus checking provided in the conventional NT file servers 24, 25, 26. Although a conventional virus checker

could be modified to run in the environment of the data mover operating system, or the data mover operating system could be modified to support a conventional virus checker, the present invention provides a way for the network file server 27 to use the virus checkers 27, 28, 29 in the NT file servers to check files in the network file server 27 in response to user access of the files in the network file server. This avoids the difficulties of porting a conventional virus checker to the network file server, and maintaining a conventional virus checker in the data mover environment of the network file server. Moreover, in many cases, the high-capacity network file server 27 is added to an existing data processing system that already includes one or more NT file servers including conventional virus checkers. In such a system, all of the files in the NT file servers 24, 25, 26 can be migrated to the high-capacity network file server 27 in order to facilitate storage management. The NT file servers 24, 25, 26 in effect become obsolete for data storage, yet they can still serve a useful function by providing virus checking services to the network file server. (Appellant's specification, page 15, line 16, to page 16, line 11.)

In general, when a client 22, 23 stores or modifies a file in the network file server 27, the network file server determines when the file needs to be scanned. When anti-virus scanning of a file has begun, other clients are blocked on any access to that file, until the scan completes on the file. The network file server 27 selects a particular one of the NT file servers 24, 25, 26 to perform the scan, in order to balance loading upon the NT file servers for anti-virus scanning processes. The virus checker in the selected NT file server performs a read-only access of the file to transfer file data from the network file server to random access memory in the selected NT

file server in order to perform the anti-virus scan in the NT file server. (Appellant's specification, page 16, lines 12-20.)

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

1. Whether claims 1-7, 9, 12-15, 18, 20-26, 28, 31, 38-39 are unpatentable under 35 U.S.C. 102(b) as being anticipated by Chen et al. (US 5,960,170).

2. Whether claim 10 is unpatenable under 35 U.S.C. 103(a) over Chen et al. in view of Cassagnol et al. (US 6,385,727 B1).

3. Whether claims 8, 11, 16, 17, 19, 27, 29-30, 32-37, and 40 are unpatentable under 35 U.S.C. 103(a) over Chen et al. in view of Cassagnol et al., Lam et al. (U.S. 5,926,636), and Tzelnic et al. (US 5,948,062).

VII. ARGUMENT

1. Claims 1-7, 9, 12-15, 18, 20-26, 28, 31, 38-39 are patentable under 35 U.S.C. 102(b) and are not anticipated by Chen et al. (US 5,960,170).

Chen et al. (FIG. 1) shows a data processing system including at least one client (300a, 300b, 300c), a LAN server 350, and a virus detection server 400. Pursuant to a request for a virus scan, a virus detection object is produced by the virus detection server and is transmitted to a client for execution. The client receives and executes the virus detection object, and the results are transmitted to the virus detection server. The virus detection server uses the results to produce an additional virus detection object which is also transmitted to the client and executed

so that the results can be transmitted to the virus detection server. The iterative production and execution of virus detection objects is continued until a determination is made as to whether the targeted file or data includes a virus. Upon a determination that a targeted file or data includes a virus, a vaccine specifically tailored to the conditions presented at the client and the type of virus detected is produced, preferably in the form of a virus treatment object. The request for a virus scan can be directly made or indirectly by a triggering event. (Abstract.) The request can be a programmed request from the client that does not require ongoing user initiation such that the scan is initiated without a request that is apparent to the user. (Col. 2, lines 3-7.) With reference to FIG. 2, in an initial step 205, a request for a virus scan is received, typically from a source external to the virus detection server 400 such as a client to be scanned. After receipt of the request, in step 210 it is determined by the virus detection server 400 whether a scan is to be performed. Preferably, a validation of the virus scan request is performed pursuant to the determination of whether a scan is to be performed. (Col. 6, lines 34-40.) FIG. 4A is a block diagram illustrating an embodiment of a virus detection server. (Col. 5, lines 1-2.) With reference to FIG. 4A, the memory of the virus detection server is preferably configured to include routines for the iterative detection of viruses. The configurations are described in further detail with reference to the iterative virus detection module 450b of FIG. 4B. (Col. 10, lines 18-30.) Referring now to FIG. 4B, an embodiment of an iterative virus detection module ("IVDM") 450b is shown to include a scanning module 454, a virus pattern module 456, a virus rules module 458, a cleaning module 460, a cleaning pattern module 462, an access managing module 464, and an access data module 466. The iterative virus detection module 450b, and its

referenced modules, includes routines for receiving virus detection requests, validating requests, producing virus detection and treatment objects, receiving the results of the execution of the virus detection and treatment objects, and using the results to produce additional virus detection and treatment objects to ultimately detect viruses and treat them. The iterative virus detection module 450b is typically implemented in software, but can also be implemented in hardware or firmware. (Col. 10, lines 52-67.)

"For a prior art reference to anticipate in terms of 35 U.S.C. § 102, every element of the claimed invention must be identically shown in a single reference." Diversitech Corp. v. Century Steps, Inc., 7 U.S.P.Q.2d 1315, 1317 (Fed. Cir. 1988), quoted in In re Bond, 15 U.S.P.Q.2d 1566, 1567 (Fed. Cir. 1990) (vacating and remanding Board holding of anticipation; the elements must be arranged in the reference as in the claim under review, although this is not an *ipsis verbis* test). See MPEP § 2131.

It is respectfully submitted that Chen et al. does not anticipate the invention of appellant's independent claims 1 and 20 because the client and servers in Chen et al. do not operate in the fashion specified in appellant's claim 1, nor are the servers programmed as specified in appellant's claim 20.

Claim 1:

In Chen et al., when a need arises to check a file in a client computer, the virus checking server sends a virus checking program to the client computer, and the client computer executes the virus checking program to check the file in the client computer. In contrast, in the method of applicant's claim 1, when a need arises to check a file in a first file server, the first file server sends the file data to the virus checking server, and the virus checking server executes the virus checking program to scan the file data in the virus checking server.

Page 15 of the Final Official Action of says: "Claim 1 recites a limitation such that "initiating the anti-virus scan of the file by sending to the second file server a request (emphasis added) for the anti-virus scan including specification of the file." Clearly the limitation cites sending a request NOT the file data as in the arguments." In reply, it is true that lines 8 to 10 of claim 1 recites "initiating the anti-virus scan of the file by sending to the second file server a request." But the fact that a request is sent to the second file server does not negate the fact that lines 11 to 14 of claim 1 recite "the second file server ... obtaining file data of the file from the first file server and storing the file data of the file into the random access memory of the second file server" which necessarily requires the file data to be sent from the first file server to the second file server. Therefore, it should be understood that in claim 1, the sending of the file data from the first file server to the second file server is performed by "the first file server ... sending to the second file server a request for the anti-virus scan including a specification of the file, and then the second file server responding to the request for the anti-virus scan of the specified file by ... obtaining file data of the file from the first file server ...." This is not done in Chen et al.

Page 15 if the Final Official Action also characterizes the operation recited in appellant's claim 1 as "intended use of the claimed invention." It is respectfully submitted that this is a mischaracterization because appellant's claim 1 is not an apparatus claim, and instead recites a method of using an apparatus

Page 15 of the Final Official Action cites In re Casey, 370 F.2d 576, 152 USPQ 235 (CCPA 1967) and In re Otto, 312 F.2d 937, 939, 136 USPQ 458, 459 (CCPA 1963). The invention of claim 1 in In re Casey was: "A taping machine comprising a supporting structure, a brush attached to said supporting structure, ..." (370 F.2d at 577.) The court said: "The claims in issue call for an apparatus or machine, vis. a tape dispensing machine. The manner or method in which such machine is to be utilized is not germane to the issue of patentability of the machine itself." (370 F.2d at 580.) The claimed invention in In re Otto also was not a method of using an apparatus. "First of all it should be remembered that the claims are directed to a particular device and a method of making that device, not to a method of curling hare wherein this particular device is used." (312 F.2d at 940.)

Claim 20:

The appellant does have apparatus claims 20 to 40, but these claims specify data processing or program storage devices (e.g. file servers) programmed in particular ways. In re Bernhart, 417 F.2d 1395, 1400, 163 U.S.P.Q. 611, 616 (C.C.P.A. 1969) ("if a machine a machine is programmed in a new and unobvious way, it is physically different from the machine without that program; its memory elements are differently arranged.") See also In re Gulack, 703 F.2d 1381,

1385, 217 U.S.P.Q. 401, 404 (Fed. Cir. 1983) ("Differences between an invention and the prior art cited against it cannot be ignored merely because those differences reside in the content of the printed matter."). Claim 20 is a system claim corresponding to claim 1, and therefore claim 20 is distinguished from Chen et al. for the same reasons as give above for claim 1.

Claims 2 and 21:

With respect to claims 2 and 21, the portion of Chen et al. (column 6, lines 34-48 and column 7, lines 4-61) cited in the Final Official Action (page 4) discloses that a client may request a virus scan and a virus detection server may validate a virus scan request to determine whether a virus scan should be performed. It is respectfully submitted, however, that Chen et al. fails to disclose that a file server containing a file determines that an anti-virus scan of the file should be performed when the client requests the file server to open the file and the file server finds that the file has not been checked for viruses.

Claims 3 and 22:

With respect to claims 3, 22, and 39, it is not seen where Chen et al. discloses that a file server determines that the anti-virus scan of the file should be performed when the client requests a file to be closed after the client writes to the file.

Claim 39:

Claim 39 combines limitations found in claims 2 and 3, and therefore distinguishes Chen et al. for the reasons given above with respect to claims 2 and 3.

Claims 5 and 24:

With respect to claims 5 and 24, it is not seen where Chen et al. discloses a file server blocks clients from accessing the file from the time that the file server determines that the anti-virus scan of the file should be performed until the anti-virus scan is completed and fails to find a virus in the file.

Claims 6 and 25:

With respect to claims 6 and 25, it is not seen where Chen et al. discloses that the first file server determines that an additional anti-virus scan of the file should not be performed in response to the access of the file by the virus checker program. Although the virus detection server 400 of Chen et al. determines that more scan of a file may and/or may not need to be performed (see Chan et al. FIG. 5), the virus detection server 400 of Chen et al. is not the same as the appellants "first file server" as recited in appellant's claims 1 or 20.

Claim 7:

With respect to claim 7, the portion of Chen et al. (column 12, line 54 through column 13, line 5) cited in the Final Official Action (page 5) relates to the indexing of virus signatures,

platform, virus type, virus identification, and virus information and criteria for determining which scanning and treatment routines to use. In contrast, the appellant's claim 7 relates to a file server maintaining in nonvolatile memory an indication of files that have not been checked for viruses, an indication of files that are in the process of being checked, and an indication of files that have been found to contain viruses. The state of a file with respect to whether the file has been or is being scanned for viruses should not be confused with criteria for determining which scanning and treatment routines to use.

Claim 9:

With respect to claim 9, the portion of Chen et al. (column 16, lines 6-17) cited in the Final Official Action (page 5) relates to the initiation of a triggering event outside of the virus checking server. In contrast, claim 9 relates to how a triggering event received by a file server is directed inside the file server to invoke a virus checker program in the file server to perform an anti-virus scan of a file. In particular, as defined in claim 9, the second file server reports a file access event to an operating system of the second file serer, and the operating system of the second file server responds by invoking the virus checker program to perform the anti-virus scan of the file.

Claim 12:

With respect to claim 12, the portion of Chen et al. (column 16, lines 15-17) cited in the Final Official Action (page 6) relates to a group of computers that a user might seek to manage

sharing a virus checking server. In contrast, the appellant's claim 12 relates to a first file server performing a load balancing procedure to select at least one of a second file server (programmed with a virus checker program) or a third file server (also programmed with a virus checker program) to perform an anti-virus scan of a file when the first file server determines that an anti-virus scan of the file should be performed.

Claim 13:

Claim 13 is distinguished from Chen et al. as stated above for claims 1, 2, 3, and 12.

Claims 14 and 18:

Claims 14 and 18 are distinguished from Chen et al. as stated above for claims 2 and 3.

Claims 15 and 28:

Claims 15 and 28 are distinguished from Chen et al. as stated above for claim 9.

Claim 26:

Claim 26 is distinguished from Chen et al. as stated above for claim 7.

Claim 31:

Claim 31 is distinguished from Chen et al. as stated above for claim 12.

Claim 38:

Claim 38 is distinguished from Chen et al. as stated above for claim 13.

2. Claim 10 is patentable under 35 U.S.C. 103(a) over Chen et al. in view of Cassagnol et al. (US 6,385,727 B1).

The policy of the Patent and Trademark Office has been to follow in each and every case the standard of patentability enunciated by the Supreme Court in Graham v. John Deere Co., 148 U.S.P.Q. 459 (1966). M.P.E.P. § 2141. As stated by the Supreme Court:

> Under § 103, the scope and content of the prior art are to be determined; differences between the prior art and the claims at issue are to be ascertained; and the level of ordinary skill in the pertinent art resolved. Against this background, the obviousness or nonobviousness of the subject matter is determined. Such secondary considerations as commercial success, long felt but unsolved needs, failure of others, etc., might be utilized to give light to the circumstances surrounding the origin of the subject matter sought to be patented. As indicia of obviousness or nonobviousness, these inquiries may have relevancy.

148 U.S.P.Q. at 467.

The problem that the inventor is trying to solve must be considered in determining whether or not the invention would have been obvious. The invention as a whole embraces the structure, properties and problems it solves. In re Wright, 848 F.2d 1216, 1219, 6 U.S.P.Q.2d 1959, 1961 (Fed. Cir. 1988).

The subject matter of the independent base claim 1 would not have been obvious from Chen et al. in view of the differences discussed above between Chen et al. and the subject matter

of claim 1, and there is nothing in Cassagnol et al. that makes up for the lack of disclosure in Chen et al. with respect to the differences discussed above. In particular, the basic operation in the appellant's system (when a need arises to check a file in a first file server, the first file server sends the file data to the virus checking server, and the virus checking server executes the virus checking program to scan the file data in the virus checking server) is entirely opposite to the basic operation in Chen et al.'s system (when a need arises to check a file in a client computer, the virus checking server sends a virus checking program to the client computer, and the client computer executes the virus checking program to check the file in the client computer). In addition, the subject matter added by the express language in the dependent claim 10 does not result from the proposed combination of Chen et al. and Cassagnol et al.

Cassagnol et al. (col. 3, lines 25-30) teaches that a processor may have a kernel mode of operation and a user mode of operation, in which non-secure software may be executed in the user mode and secure software may be executed in the kernel mode. However, this teaching is too general to suggest the specific construction defined in appellant's claim 10. In particular, it is not seen how the proposed combination of Chen et al. and Cassagnol et al. would provide the appellant's server for virus checking executing in the user mode that receives the request for the anti-virus scan from the first file server and forwards the request to a virus checker initiator driver executing in the kernel mode, and the virus checker initiator driver executing in the kernel mode initiates a file access event, and the virus checker program initiates the anti-virus scan of the file in response to the virus checker initiator driver initiating the file access event. (See appellant's FIG. 3 and appellant's specification, page 19 line 4 to page 20 line 10.)

Where the prior art references fail to teach a claim limitation, there must be "concrete evidence" in the record to support an obviousness rejection. "Basic knowledge" or "common sense" is insufficient. In re Zurko, 258 F.3d 1379, 1385-86, 59 U.S.P.Q.2d 1693, 1697 (Fed. Cir. 2001); In re Gordon et al., 733 F.2d 900, 902, 221 U.S.P.Q. 1125, 1127 (Fed. Cir. 1984) (mere fact that prior art could be modified by turning apparatus upside down does not make modification obvious unless prior art suggests desirability of modification); Ex Parte Kaiser, 194 U.S.P.Q. 47, 48 (PTO Bd. of Appeals 1975) (Examiner's failure to indicate anywhere in the record his reason for finding alteration of reference to be obvious militates against rejection).

3. Claims 8, 11, 16, 17, 19, 27, 29-30, 32-37, and 40 are patentable under 35 U.S.C. 103(a) over Chen et al. in view of Cassagnol et al., Lam et al., (U.S. 5,926,636), and Tzelnic et al. (US 5,948,062).

The subject matter of the independent base claims would not have been obvious from Chen et al. in view of the differences discussed above between Chen et al. and the subject matter of the independent base claims, and there is nothing in Cassagnol et al., Lam et al., and Tzelnic et al. that makes up for the lack of disclosure in Chen et al. with respect to the differences discussed above. In particular, the basic operation in the appellant's system (when a need arises to check a file in a first file server, the first file server sends the file data to the virus checking server, and the virus checking server executes the virus checking program to scan the file data in the virus checking server) is entirely opposite to the basic operation in Chen et al.'s system (when a need arises to check a file in a client computer, the virus checking server sends a virus

checking program to the client computer, and the client computer executes the virus checking program to check the file in the client computer). Where the prior art references fail to teach a claim limitation, there must be "concrete evidence" in the record to support an obviousness rejection. "Basic knowledge" or "common sense" is insufficient. In re Zurko, 258 F.3d 1379, 1385-86, 59 U.S.P.Q.2d 1693, 1697 (Fed. Cir. 2001); In re Gordon et al., 733 F.2d 900, 902, 221 U.S.P.Q. 1125, 1127 (Fed. Cir. 1984) (mere fact that prior art could be modified by turning apparatus upside down does not make modification obvious unless prior art suggests desirability of modification); Ex Parte Kaiser, 194 U.S.P.Q. 47, 48 (PTO Bd. of Appeals 1975) (Examiner's failure to indicate anywhere in the record his reason for finding alteration of reference to be obvious militates against rejection).

As introduced above in the summary of the invention, the appellants' invention provides advantages not evident from the cited references. The present invention provides a way for the network file server 27 to use the virus checkers 27, 28, 29 in the NT file servers to check files in the network file server 27 in response to user access of the files in the network file server. This avoids the difficulties of porting a conventional virus checker to the network file server, and maintaining a conventional virus checker in the data mover environment of the network file server. Moreover, in many cases, the high-capacity network file server 27 is added to an existing data processing system that already includes one or more NT file servers including conventional virus checkers. In such a system, all of the files in the NT file servers 24, 25, 26 can be migrated to the high-capacity network file server 27 in order to facilitate storage management. The NT file servers 24, 25, 26 in effect become obsolete for data storage, yet they can still serve a useful

function by providing virus checking services to the network file server. (Appellant's specification, page 15, line 16, to page 16, line 11.)

Claim 11:

With respect to appellant's claim 11, it is not seen where the memory 414 in FIG. 4A of Chen et al. includes an "input/output manager in the operating system". The routines for the iterative detection of viruses would not be in the operating system. In addition, it is not understood how the server component management API of Lam et al., which rejects a remote procedure call for an incompatible version, would suggest the appellant's "input/output manager in the operating system of the second file server receives a file access call from the virus checker initiator driver, and responds by directing a report of the file access event to the virus checker program." The appellant's virus checker initiator driver and the appellant's input/output manager are directed to solving the problem of how to stimulate the operating system of the NT file server to cause the conventional virus checker program to check an external file. Therefore, if the conventional virus checker program were upgraded in a fashion compatible with the Windows NT/2000 operating system, the upgraded virus checker program would continue to be invoked by the RPC client for virus checking in the network file server. (See appellant's FIG. 3 and appellant's specification, page 19 line 4 to page 20 line 10.) The cited portion of Lam et al. would not have suggested a solution to this problem because Lam et al. deals with rejecting a remote procedure call for an incompatible version.

Claim 16:

With respect to claim 16, it is respectfully submitted that Chen et al. is distinguished as stated above with respect to claim 1 to the extent that claim 16 calls for "a virus checker program executing in the second server in the user mode, and the virus checker program responds by obtaining file data from the file in the first server and storing the file data in random access memory in the second server, and performing an anti-virus scan upon the file data in the random access memory in the second server." Chen et al. is also distinguished as stated above with respect to claim 11 with reference to the recitation in claim 16 of the operating system of the second server including an input/output manager executing in the kernel mode. In view of the statement in the Final Official Action (page 11) that Chen does not explicitly mention (1) processes executing in a user mode and processes executing in a kernel mode, and (2) the role of the input/output manager, it is respectfully submitted that the cited portions of Chen et al. (col. 10, lines 20-23 and 20-23) fail to disclose "the server for virus checking forwards the request to a virus checker initiator driver executing in the second server in the kernel mode, and the virus checker initiator driver responds to receipt of the request by sending a file access call to the input/output manager." The Final Official Action cites Cassagnol and Lam for these features. Cassagnol (col. 3, lines 25-30) teaches that a processor may have a kernel mode of operation and a user mode of operation, in which non-secure software may be executed in the user mode and secure software may be executed in the kernel mode. However, this teaching is too general to suggest the specific construction defined in appellant's claim 16. Lam et al. is distinguished as stated above with reference to claim 11. It is not understood how the server component

management API of Lam et al., which rejects a remote procedure call for an incompatible version, would have suggested both the appellant's virus checker initiator driver and the appellant's input output manager, and would have also suggested the appellant's virus checker initiator driver executing in the second server in the kernel mode, and the virus checker initiator driver responds to receipt of the request by sending a file access call to the input/output manager. Thus, it is respectfully submitted that it would not have been obvious to reconstruct the subject matter of appellant's claim 16 by picking and choosing various things from Chen et al., Cassagnol et al., and Lam et al., adding the missing elements, and modifying that combination as proposed in the Final Official Action.

Hindsight reconstruction, using the appellant's specification itself as a guide, is improper because it fails to consider the subject matter of the invention "as a whole" and fails to consider the invention as of the date at which the invention was made. "[T]here must be some motivation, suggestion, or teaching of the desirability of making the specific combination that was made by the applicant." In re Lee, 277 F.3d 1338, 1343, 61 U.S.P.Q.2d 1430, 1435 (Fed. Cir. 2002) (quoting In re Dance, 160 F.3d 1339, 1343, 48 U.S.P.Q.2d 1635, 1637 (Fed. Cir. 1998)). "[T]eachings of references can be combined only if there is some suggestion or incentive to do so." In re Fine, 837 F.2d 1071, 1075, 5 U.S.P.Q.2d 1596, 1600 (Fed. Cir. 1988) (Emphasis in original) (quoting ACS Hosp. Sys., Inc. v. Montefiore Hosp., 732 F.2d 1572, 1577, 221 U.S.P.Q. 929, 933 (Fed. Cir. 1984)). "[P]articular findings must be made as to the reason the skilled artisan, with no knowledge of the claimed invention, would have selected these components for combination in the manner claimed." In re Kotzab, 217 F.3d 1365, 1371, 55 U.S.P.Q.2d 1313,

1317 (Fed. Cir. 2000). See, for example, Fromson v. Advance Offset Plate, Inc., 755 F.2d 1549, 1556, 225 U.S.P.Q. 26, 31 (Fed. Cir. 1985) (nothing of record plainly indicated that it would have been obvious to combine previously separate lithography steps into one process).

Claim 19:

With respect to appellant's claim 19, see the appellant's remarks above with respect to claim 16.

Claim 27:

With respect to appellant's claim 27, see the appellant's remarks above with respect to claim 8.

Claim 29:

With respect to appellant's claim 29, see the appellant's remarks above with respect to claim 10.

Claim 30:

With respect to appellant's claim 30, see the appellant's remarks above with respect to claim 11.

Claims 34 and 37:

With respect to appellant's claims 34 and 37, see the appellant's remarks above with respect to claims 1, 16, and 20.

Claim 40:

With respect to claim 40, see the appellant's remarks above with respect to claim 16.

Claim 17:

With respect to claim 17, see the appellant's remarks above with respect to claim 1, 2, 3, and 12.

Claim 32:

With respect to claim 32, see the appellant's remarks above with respect to claims 1, 2, 3, and 12.

Claims 33 and 36:

With respect to claims 33 and 36, see the appellant's remarks above with respect to claims 2 and 3.

Claim 35:

With respect to claim 35, see the appellant's remarks above with respect to claims 1, 2, 3, and 12.

In view of the above, the rejection of the claims should be reversed.

14 Sept. 2005

Respectfully submitted,

Richard C. Auchterlonie
Reg. No. 30,607

NOVAK DRUCE & QUIGG, LLP
1000 Louisiana, Suite 5320
Houston, TX 77002
713-751-0655

VIII. CLAIMS APPENDIX

The claims involved in this appeal are as follows:

1. In a data processing system including at least one client, a first file server coupled to the client for data access of the client to at least one file in the first file server, and at least a second file server coupled to the first file server for data access of the second file server to the file in the first file server, the second file server being programmed with a virus checker program, the virus checker program being executable by the second file server to perform an anti-virus scan upon file data in random access memory of the second file server, a method comprising:

the first file server responding to a request from the client for access to the file in the first file server by determining that an anti-virus scan of the file should be performed, and initiating the anti-virus scan of the file by sending to the second file server a request for the anti-virus scan including a specification of the file, and then

the second file server responding to the request for the anti-virus scan by invoking the virus checker program to perform an anti-virus scan of the specified file by obtaining file data of the file from the first file server and storing the file data of the file into the random access memory of the second file server and performing the anti-virus scan upon the file data of the file in the random access memory.

2. The method as claimed in claim 1, wherein the first file server determines that the anti-virus scan of the file should be performed when the client requests the first file server to open the file and the first file server finds that the file has not been checked for viruses.

3. The method as claimed in claim 1, wherein the first file server determines that the anti-virus scan of the file should be performed when the client requests a file to be closed after the client writes to the file.

4. The method as claimed in claim 1, wherein the first file server applies a filter to a file extension of the file to determine that the anti-virus scan of the file should be performed.

5. The method as claimed in claim 1, wherein the first file server blocks clients from accessing the file from the time that the first file server determines that the anti-virus scan of the file should be performed until the anti-virus scan is completed and fails to find a virus in the file.

6. The method as claimed in claim 1, wherein the first file server determines that an additional anti-virus scan of the file should not be performed in response to the access of the file by the virus checker program.

7. The method as claimed in claim 1, wherein the first file server maintains in nonvolatile memory an indication of files that have not been checked for viruses, an indication of files that

are in the process of being checked, and an indication of files that have been found to contain viruses.

8. The method as claimed in claim 1, wherein the request for the anti-virus scan including a specification of the file is an Open Network Computing Remote Procedure Call.

9. The method as claimed in claim 1, wherein the second file server receives the request for the anti-virus scan and indirectly invokes the virus checker program by reporting a file access event to an operating system of the second file server, and the operating system of the second file server responds by invoking the virus checker program to perform the anti-virus scan of the file.

10. The method as claimed in claim 1, wherein the operating system of the second file server supports processes executing in a user mode and processes executing in a kernel mode, and a server for virus checking executing in the user mode receives the request for the anti-virus scan from the first file server and forwards the request to a virus checker initiator driver executing in the kernel mode, and the virus checker initiator driver executing in the kernel mode initiates a file access event, and the virus checker program initiates the anti-virus scan of the file in response to the virus checker initiator driver initiating the file access event.

11. The method as claimed in claim 10, wherein an input/output manager in the operating system of the second file server receives a file access call from the virus checker initiator driver, and responds by directing a report of the file access event to the virus checker program.

12. The method as claimed in claim 1, wherein the data processing system includes at least a third file server coupled to the first file server for data access of the third file server to the file in the first file server, the third file server also being programmed with a virus checker program that is executable by the third file server to perform an anti-virus scan upon file data in random access memory of the third file server, wherein the first file server performs a load balancing procedure to select one of at least the second file server or the third file server to perform an anti-virus scan of the file when the first file server determines that an anti-virus scan of the file should be performed.

13. A method of operating a network file server to initiate a virus scan upon a file stored in the network file server, the network file server being coupled to at least one client for access of the client to at least one file in the network file server, the network file server being coupled to a plurality of secondary servers for access of the secondary servers to the file stored in the network file server, the network file server including a cached disk array and a plurality of data mover computers coupled to the data network and coupled to the cached disk array for responding to client requests for data access to storage in the cached disk array, each secondary server being

programmed with a virus checker program executable for performing an anti-virus scan upon file data in random access memory of said each secondary server, a method comprising:

at least one of the data movers in the network file server responding to a request from the client for access to the file in the network file server by applying a filter upon a file extension of the file upon opening or closing of the file to determine that an anti-virus scan of the file should be performed, and initiating the anti-virus scan of the file by applying a load balancing procedure for selecting one of the secondary servers for performing the anti-virus scan of the file, and sending to the selected secondary server a request for the anti-virus scan including a specification of the file, and then

the selected secondary server responding to the request for the anti-virus scan by invoking the virus checker program in the selected secondary server to perform an anti-virus scan of the specified file by obtaining file data of the file from the network file server and storing the file data of the file into the random access memory of the selected secondary server and performing the anti-virus scan upon the file data of the file in the random access memory of the selected secondary server.

14. The method as claimed in claim 13, wherein the file is an executable file, and the network file server determines that the anti-virus scan of the file should be performed when the client requests the network file server to open the file and the network file server finds that the file has not been checked for viruses, and the network file server also determines that the anti-

virus scan of the file should be performed when the client requests the file to be closed after the client writes to the file.

15. The method as claimed in claim 13, wherein the selected secondary server receives the request for the anti-virus scan and indirectly invokes the virus checker program by causing an operating system of the selected secondary server to invoke the virus checker program in the selected secondary server to perform the anti-virus scan of the file.

16. In a data network including a first server and a second server, the second server being coupled by a data network to the first server for access of the second server to at least one file stored in the first server, wherein the second server is programmed with an operating system supporting processes executing in a user mode and processes executing in a kernel mode, the operating system of the second server including an input/output manager executing in the kernel mode, a method of operating the second server to perform an anti-virus scan upon the file in the first server, said method comprising:

a server for virus checking executing in the second server in the user mode receives from the network a request for the anti-virus scan upon the file, and then

the server for virus checking forwards the request to a virus checker initiator driver executing in the second server in the kernel mode, and the virus checker initiator driver responds to receipt of the request by sending a file access call to the input/output manager, and then

the input/output manager responds to the file access call by directing a report of a file access event to a virus checker program executing in the second server in the user mode, and the virus checker program responds by obtaining file data from the file in the first server and storing the file data in random access memory in the second server, and performing an anti-virus scan upon the file data in the random access memory in the second server.

17. In a data processing system including at least one client, at least one network file server coupled to the client by a data network for access of the client to at least one file in the network file server, and a plurality of NT file servers coupled to the network file server by the data network for data access of the NT file servers to the file in the network file server, the network file server including a cached disk array and a plurality of data mover computers coupled to the data network and coupled to the cached disk array for responding to client requests for data access to storage in the cached disk array, each of the NT file servers being programmed with a virus checker program, the virus checker program in each NT file server being executable by said each NT file server to perform an anti-virus scan upon file data in random access memory of said each NT file server, a method comprising:

a data mover in the network file server responding to a request from the client for access to the file in the network file server by applying a filter upon a file extension of the file upon opening or closing of the file to determine that an anti-virus scan of the file should be performed, and initiating the anti-virus scan of the file by selecting a next one of the NT file servers in

round-robin fashion and sending to the selected NT file server a request for the anti-virus scan including a specification of the file, and then

the selected NT file server responding to the request for the anti-virus scan by invoking the virus checker program in the selected NT file server to perform an anti-virus scan of the specified file by obtaining file data of the file from the network file server and storing the file data of the file in the random access memory of the selected NT file server and performing the anti-virus scan upon the file data of the file in the random access memory of the selected NT file server.

18. The method as claimed in claim 17, wherein the file is an executable file and the network file server determines that the anti-virus scan of the file should be performed when the client requests the network file server to open the file and the network file server finds that the file has not been checked for viruses, and the network file server also determines that the anti-virus scan of the file should be performed when the client requests the file to be closed after the client writes to the file.

19. The method as claimed in claim 17, wherein the operating system of the selected NT file server supports processes executing in a user mode and processes executing in a kernel mode, a server for virus checking executing in the user mode receives the request for the anti-virus scan from the network file server and forwards the request to a virus checker initiator driver executing in the kernel mode, the virus checker initiator driver executing in the kernel mode sends a file

access call to an input/output manager in the operating system of the selected NT file server, the input/output manager responds to the file access call by directing a report of a file access event to the virus checker program in the selected NT file server, and the virus checker program in the selected NT file server initiates the anti-virus scan of the file in response to receiving the report of the file access event from the input/output manager.

20. A data processing system comprising:

at least one client;

a first file server coupled to the client for access of the client to at least one file in the first file server; and

at least a second file server coupled to the first file server for data access of the second file server to the file in the first file server, the second file server being programmed with a virus checker program, the virus checker program being executable by the second file server to perform an anti-virus scan upon file data in random access memory of the second file server;

wherein the first file server is programmed to respond to a request from the client for access to the file in the first file server by determining that an anti-virus scan of the file should be performed, and initiating the anti-virus scan of the file by sending to the second file server a request for the anti-virus scan including a specification of the file; and

the second file server is programmed to respond to the request for the anti-virus scan by invoking the virus checker program to perform an anti-virus scan of the specified file by obtaining file data of the file from the first file server and storing the file data of the file in the

random access memory of the second file server and performing the anti-virus scan upon the file data in the random access memory.

21. The data processing system as claimed in claim 20, wherein the first file server is programmed to determine that the anti-virus scan of the file should be performed when the client requests the first file server to open the file and the first file server finds that the file has not been checked for viruses.

22. The data processing system as claimed in claim 20, wherein the first file server is programmed to determine that the anti-virus scan of the file should be performed when the client requests a file to be closed after the client writes to the file.

23. The data processing system as claimed in claim 20, wherein the first file server is programmed to apply a filter to a file extension of the file to determine that the anti-virus scan of the file should be performed.

24. The data processing system as claimed in claim 20, wherein the first file server is programmed to block clients from accessing the file from the time that the first file server determines that the anti-virus scan of the file should be performed until the anti-virus scan is completed and fails to find a virus in the file.

25. The data processing system as claimed in claim 20, wherein the first file server is programmed to determine that an additional anti-virus scan of the file should not be performed in response to the access of the file by the virus checker program.

26. The data processing system as claimed in claim 20, wherein the first file server is programmed to maintain in nonvolatile memory an indication of files that have not been checked for viruses, an indication of files that are in the process of being checked, and an indication of files that have been found to contain viruses.

27. The data processing system as claimed in claim 20, wherein the request for the anti-virus scan including a specification of the file is an Open Network Computing Remote Procedure Call.

28. The data processing system as claimed in claim 20, wherein the second file server is programmed to receive the request for the anti-virus scan and indirectly invoke the virus checker program by causing an operating system of the second file server to invoke the virus checker program to perform the anti-virus scan of the file.

29. The data processing system as claimed in claim 20, wherein the operating system of the second file server supports processes executing in a user mode and processes executing in a kernel mode, and the second file server includes a server for virus checking that is executable in the user mode and a virus checker initiator driver that is executable in the kernel mode, the server

for virus checking being executable for receiving the request for the anti-virus scan from the first file server and forwarding the request to the virus checker initiator driver, and the virus checker initiator driver is executable for causing the operating system reporting a file access event to the virus checker program, and the virus checker program is executable for initiating the anti-virus scan of the file in response to the report of the file access event.

30. The data processing system as claimed in claim 29, wherein the operating system of the second file server includes an input/output manager that is executable in the kernel mode for receiving a file access call from the virus checker initiator driver, and responding to the file access call by directing the report of the file access event to the virus checker program.

31. The data processing system as claimed in claim 20, which further includes at least a third file server coupled to the first file server for data access of the third file server to the file in the first file server, the third file server also being programmed with a virus checker program that is executable by the third file server to perform an anti-virus scan upon file data in random access memory of the third file server, and wherein the first file server is programmed for performing a load balancing procedure to select one of at least the second file server or the third file server to perform an anti-virus scan of the file when the first file server determines that an anti-virus scan of the file should be performed.

32. A network file server adapted for coupling to at least one client for access of the client to at least one file in the network file server, the network file server also being adapted for coupling to a plurality of secondary servers for access of the secondary servers to the file stored in the network file server, each secondary server being programmed with a virus checker program executable for transferring file data from the file in the network file server to random access memory in said each secondary server, and performing an anti-virus scan upon the file data in the random access memory of said each secondary server, the network file server comprising:

a cached disk array; and

a plurality of data mover computers coupled to the data network and coupled to the cached disk array for responding to client requests for data access to storage in the cached disk array, wherein at least one of the data movers is programmed to respond to a request from the client for access to the file in the network file server by applying a filter upon a file extension of the file upon opening or closing of the file to determine that an anti-virus scan of the file should be performed, and initiating the anti-virus scan of the file by applying a load balancing procedure for selecting one of the secondary servers for performing the anti-virus scan of the file, and sending to the selected secondary server a request for the anti-virus scan including a specification of the file.

33. The data processing system as claimed in claim 32, wherein the file is executable and the network file server is programmed to determine that an anti-virus scan of the file should be performed when the client requests the network file server to open the file and the network file

server finds that the file has not been checked for viruses, and the network file server is also programmed to determine that the anti-virus scan of the file should be performed when the client requests the file to be closed after the client writes to the file.

34. A secondary server adapted for coupling to a primary server in a data network for access to data in files in the primary server, wherein the secondary server is programmed with an operating system supporting processes executing in a user mode and processes executing in a kernel mode, the operating system including an input/output manager executable in the kernel mode, wherein the secondary server is further programmed with:

a server for virus checking executable in the user mode;

a virus checking driver executable in the kernel mode; and

a virus checker program executable in the user mode;

wherein the server for virus checking is executable for receiving from the network a request for an anti-virus scan upon a specified file in the primary server, and for forwarding the request to the virus checker initiator driver;

wherein the virus checker initiator driver is executable for responding to receipt of the request from the server for virus checking by sending a file access call upon the specified file to the input/output manager;

wherein the input/output manager is executable for responding to the file access call by directing a report of a file access event upon the specified file to the virus checker program; and

wherein the virus checker program is executable for responding to the report of the file access event by transferring file data from the specified file in the primary server to random access memory in the secondary server, and performing an anti-virus scan upon the file data in the random access memory in the secondary server.

35. A data processing system comprising:

at least one client;

at least one network file server coupled to the client by a data network for access of the client to at least one file in the network file server; and

a plurality of NT file servers coupled to the network file server by the data network for data access of the NT file servers to the file in the network file server;

wherein the network file server includes a cached disk array and a plurality of data mover computers coupled to the data network and coupled to the cached disk array for responding to client requests for data access to storage in the cached disk array;

wherein each of the NT file servers is programmed with a virus checker program, the virus checker program being executable by said each NT file server to perform an anti-virus scan upon file data in random access memory of said each NT file server;

wherein at least one data mover in the network file server is programmed for responding to a request from the client for access to the file in the network file server by applying a filter upon a file extension of the file upon opening or closing of the file to determine that an anti-virus scan of the file should be performed, and initiating the anti-virus scan of the file by selecting a

next one of the NT file servers in round-robin fashion and sending to the selected NT file server a request for the anti-virus scan including a specification of the file, and

wherein each NT file server is programmed to respond to the request for the anti-virus scan by invoking the virus checker program in said each NT file server to perform an anti-virus scan of the specified file by obtaining file data of the specified file from the network file server and storing the file data of the specified file in the random access memory of said each NT file server and performing the anti-virus scan upon the file data of the specified file in the random access memory of said each NT file server.

36. The data processing system as claimed in claim 35, wherein the file is an executable file, and the network file server is programmed for determining that the anti-virus scan of the file should be performed when the client requests the network file server to open the file and the network file server finds that the file has not been checked for viruses, and the network file server also determines that the anti-virus scan of the file should be performed when the client requests the file to be closed after the client writes to the file.

37. The data processing system as claimed in claim 35, wherein the selected NT file server is programmed with:

an operating system supporting processes executing in a user mode and processes executing in a kernel mode, the operating system including an input/output manager executable in the kernel mode;

a server for virus checking executable in the user mode; and

a virus checker initiator driver executable in the kernel mode;

wherein the server for virus checking is executable for receiving the request for the anti-virus scan from the network file server and forwarding the request to the virus checker initiator driver;

wherein the virus checker initiator driver is executable for responding to the request from the server for virus checking by sending a file access call to the input/output manager;

wherein the input/output manager is executable for responding to the file access call by directing a report of a file access event to the virus checker program in the selected NT file server; and

wherein the virus checker program in the selected NT file server is executable for performing the anti-virus scan of the file in response to receiving the report of the file access event from the input/output manager.

38. A program storage device containing a program executable by a network file server, the network file server being adapted for coupling to at least one client for access of the client to at least one file in the network file server, the network file server also being adapted for coupling to a plurality of secondary servers for access of the secondary servers to the file stored in the network file server, each secondary server being programmed with a virus checker program executable for transferring file data from the file in the network file server to random access memory in said each secondary server, and performing an anti-virus scan upon the file data in

the random access memory of said each secondary server, the program contained in the program storage device being executable by the network file server for responding to a request from the client for access to the file in the network file server by applying a filter upon a file extension of the file upon opening or closing of the file to determine that an anti-virus scan of the file should be performed, and initiating the anti-virus scan of the file by applying a load balancing procedure for selecting one of the secondary servers for performing the anti-virus scan of the file, and sending to the selected secondary server a request for the anti-virus scan including a specification of the file.

39. The program storage device as claimed in claim 38, wherein the file is an executable file, and the program contained in the program storage device is also executable by the network file server for determining that an anti-virus scan of the file should be performed when the client requests the network file server to open the file and the network file server finds that the file has not been checked for viruses, and the program contained in the program storage device is also executable by the network file server for determining that the anti-virus scan of the file should be performed when the client requests the file to be closed after the client writes to the file.

40. A program storage device containing a program executable by a secondary server, the secondary server being adapted for coupling to a primary server in a data network for access to data in files in the primary server, wherein the secondary server is programmable with an operating system supporting processes executing in a user mode and processes executing in a

kernel mode, the operating system including an input/output manager executable in the kernel mode, and the secondary server is also programmable with a virus checker program for performing an anti-virus scan upon file data in response to a file opening event being reported to the input/output manager, wherein the program contained in the program storage device includes:

a server for virus checking executable in the user mode; and

a virus checking driver executable in the kernel mode;

wherein the server for virus checking is executable for receiving from the network a request for the anti-virus scan upon a specified file in the primary server, and for forwarding the request to the virus checker initiator driver; and

wherein the virus checker initiator driver is executable for responding to receipt of the request from the server for virus checking by sending a file access call upon the specified file to the input/output manager, whereby the input/output manager directs a report of a file access event upon the specified file to the virus checker program to initiate an anti-virus scan upon file data of the specified file.

PTO/SB/21 (09-04)
Approved for use through 07/31/2006. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

TRANSMITTAL FORM

(to be used for all correspondence after initial filing)

Total Number of Pages in This Submission: 45

Application Number	09/804,320
Filing Date	03/12/2001
First Named Inventor	Frank S. Caccavale
Art Unit	2135
Examiner Name	Truong, Thanhnga B
Attorney Docket Number	10830.075.NPUS0

ENCLOSURES *(Check all that apply)*

- ☐ Fee Transmittal Form
  - ☐ Fee Attached
- ☐ Amendment/Reply
  - ☐ After Final
  - ☐ Affidavits/declaration(s)
- ☐ Extension of Time Request
- ☐ Express Abandonment Request
- ☐ Information Disclosure Statement
- ☐ Certified Copy of Priority Document(s)
- ☐ Reply to Missing Parts/ Incomplete Application
  - ☐ Reply to Missing Parts under 37 CFR 1.52 or 1.53
- ☐ Drawing(s)
- ☐ Licensing-related Papers
- ☐ Petition
- ☐ Petition to Convert to a Provisional Application
- ☐ Power of Attorney, Revocation Change of Correspondence Address
- ☐ Terminal Disclaimer
- ☐ Request for Refund
- ☐ CD, Number of CD(s) _____
  - ☐ Landscape Table on CD
- ☐ After Allowance Communication to TC
- ☐ Appeal Communication to Board of Appeals and Interferences
- ☑ Appeal Communication to TC **(Appeal Notice, Brief, Reply Brief)**
- ☐ Proprietary Information
- ☐ Status Letter
- ☑ Other Enclosure(s) (please Identify below):

Return Post Card
Fee Transmittal Form

Remarks

SIGNATURE OF APPLICANT, ATTORNEY, OR AGENT

Firm Name	Novak Druce & Quigg, LLP		
Signature	Richard C. Auchterlonie		
Printed name	Richard C. Auchterlonie		
Date	14 Sept. 2005	Reg. No.	30,607

CERTIFICATE OF TRANSMISSION/MAILING

I hereby certify that this correspondence is being facsimile transmitted to the USPTO or deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to: Commissioner for Patents. P.O. Box 1450. Alexandria, VA 22313-1450 on the date shown below: EXPRESS MAIL MAILING LABEL NO. ED802269137US

Signature	Richard C. Auchterlonie		
Typed or printed name	Richard C. Auchterlonie, Reg. 30,607	Date	14 Sept. 2005

PTO/SB/17 (10-04)
Approved for use through 07/31/2006. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

FEE TRANSMITTAL for FY 2005

Effective 10/01/2004. Patent fees are subject to annual revision.

☐ Applicant claims small entity status. See 37 CFR 1.27

TOTAL AMOUNT OF PAYMENT ($) 500.

Complete if Known	
Application Number	09/804,320
Filing Date	03/12/2001
First Named Inventor	Caccavale, Frank S.
Examiner Name	Truong, Thanhnga B.
Art Unit	2135
Attorney Docket No.	10830.0075.NPUS00

METHOD OF PAYMENT *(check all that apply)*

☐ Check ☐ Credit card ☐ Money Order ☐ Other ☐ None

☑ Deposit Account:

Deposit Account Number: 05-0889

Deposit Account Name: EMC Corporation

The Director is authorized to: *(check all that apply)*

☑ Charge fee(s) indicated below ☑ Credit any overpayments

☑ Charge any additional fee(s) or any underpayment of fee(s)

☐ Charge fee(s) indicated below, **except for the filing fee** to the above-identified deposit account.

FEE CALCULATION

1. BASIC FILING FEE

Large Entity Fee Code	Large Entity Fee ($)	Small Entity Fee Code	Small Entity Fee ($)	Fee Description	Fee Paid
1001	790	2001	395	Utility filing fee	
1002	350	2002	175	Design filing fee	
1003	550	2003	275	Plant filing fee	
1004	790	2004	395	Reissue filing fee	
1005	160	2005	80	Provisional filing fee	

SUBTOTAL (1) ($)

2. EXTRA CLAIM FEES FOR UTILITY AND REISSUE

		Extra Claims		Fee from below	Fee Paid
Total Claims		-20** =		X	=
Independent Claims		- 3** =		X	=
Multiple Dependent					=

Large Entity Fee Code	Large Entity Fee ($)	Small Entity Fee Code	Small Entity Fee ($)	Fee Description
1202	18	2202	9	Claims in excess of 20
1201	88	2201	44	Independent claims in excess of 3
1203	300	2203	150	Multiple dependent claim, if not paid
1204	88	2204	44	** Reissue independent claims over original patent
1205	18	2205	9	** Reissue claims in excess of 20 and over original patent

SUBTOTAL (2) ($)

***or number previously paid, if greater; For Reissues, see above*

FEE CALCULATION (continued)

3. ADDITIONAL FEES

Large Entity Fee Code	Large Entity Fee ($)	Small Entity Fee Code	Small Entity Fee ($)	Fee Description	Fee Paid
1051	130	2051	65	Surcharge - late filing fee or oath	
1052	50	2052	25	Surcharge - late provisional filing fee or cover sheet	
1053	130	1053	130	Non-English specification	
1812	2,520	1812	2,520	For filing a request for *ex parte* reexamination	
1804	920*	1804	920*	Requesting publication of SIR prior to Examiner action	
1805	1,840*	1805	1,840*	Requesting publication of SIR after Examiner action	
1251	110	2251	55	Extension for reply within first month	
1252	430	2252	215	Extension for reply within second month	
1253	980	2253	490	Extension for reply within third month	
1254	1,530	2254	765	Extension for reply within fourth month	
1255	2,080	2255	1,040	Extension for reply within fifth month	
1401	340	2401	170	Notice of Appeal	
1402	340	2402	170	Filing a brief in support of an appeal	500
1403	300	2403	150	Request for oral hearing	
1451	1,510	1451	1,510	Petition to institute a public use proceeding	
1452	110	2452	55	Petition to revive - unavoidable	
1453	1,330	2453	665	Petition to revive - unintentional	
1501	1,370	2501	685	Utility issue fee (or reissue)	
1502	490	2502	245	Design issue fee	
1503	660	2503	330	Plant issue fee	
1460	130	1460	130	Petitions to the Commissioner	
1807	50	1807	50	Processing fee under 37 CFR 1.17(q)	
1806	180	1806	180	Submission of Information Disclosure Stmt	
8021	40	8021	40	Recording each patent assignment per property (times number of properties)	
1809	790	2809	395	Filing a submission after final rejection (37 CFR 1.129(a))	
1810	790	2810	395	For each additional invention to be examined (37 CFR 1.129(b))	
1801	790	2801	395	Request for Continued Examination (RCE)	
1802	900	1802	900	Request for expedited examination of a design application	

Other fee (specify) _____

*Reduced by Basic Filing Fee Paid

SUBTOTAL (3) ($) 500

SUBMITTED BY				(Complete *(if applicable)*)	
Name (Print/Type)	Richard C. Auchterlonie	*Registration No. (Attorney/Agent)*	30,607	*Telephone*	713-751-0655
Signature	Richard C. Auchterlonie			*Date*	14 Sept. 2005

WARNING: Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2038.

This collection of information is required by 37 CFR 1.17 and 1.27. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.